WARWICK
THE UNIVERSITY OF WARWICK

Manuscript version: Author's Accepted Manuscript
The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:
http://wrap.warwick.ac.uk/136448

How to cite:
Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:
Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

warwick.ac.uk/lib-publications

Deep Neural Learning Based Distributed Predictive Control for Offshore Wind Farm Using High Fidelity LES Data

Xiuxing Yin and Xiaowei Zhao

***Abstract*—The paper explores the deep neural learning (DNL) based predictive control approach for offshore wind farm using high fidelity large eddy simulations (LES) data. The DNL architecture is defined by combining the Long Short-Term Memory (LSTM) units with Convolutional Neural Networks (CNN) for feature extraction and prediction of the offshore wind farm. This hybrid CNN-LSTM model is developed based on the dynamic models of the wind farm and wind turbines as well as higher-fidelity LES data. Then, distributed and decentralized model predictive control (MPC) methods are developed based on the hybrid model for maximizing the wind farm power generation and minimizing the usage of the control commands. Extensive simulations based on a two-turbine and a nine-turbine wind farm cases demonstrate the high prediction accuracy (97% or more) of the trained CNN-LSTM models. They also show that the distributed MPC can achieve up to 38% increase in power generation at farm scale than the decentralized MPC. The computational time of the distributed MPC is around 0.7s at each time step, which is sufficiently fast as a real-time control solution to wind farm operations.**

***Index Terms*—Deep neural learning; Offshore wind farm; Model predictive control; LES data.**

I. Introduction

THE European Union has set an ambitious target that 20% of the energy consumed in Europe should be contributed from renewables by 2020. Offshore wind plays a leading role in achieving this target [1]. However, the power generation efficiency of the offshore wind farm is currently still not very attractive, and an improvement is required, which can further reduce offshore wind power costs. Control systems are seen as an important enabler in maximizing wind energy capture. Common industrial practice in offshore wind farm operation is to control each turbine individually using locally available measurements, which causes the whole wind farm to operate in a non-optimum way. Actually, wind turbines in a wind farm experience extensive wake interactions which reduce energy extraction and increase dynamic mechanical loads. Neglecting the wake interactions will result in great suboptimal performance of wind farm operations. Thus, it is necessary to develop an efficient wind farm control strategy to coordinate the turbine operations at farm scale to optimize the overall operations in both energy production and maintenance.

The wind farm control can be exploited by leveraging farm-level interactions between wind turbines and has been receiving an increasing amount of attention. In [2], the feasibility of the Bayesian Ascent (BA) algorithm was explored for the optimal coordinated control actions of the wind turbines within a farm, using limited amount of data. In [3], a wind farm controller was designed with both local and central levels of control. In [4], the optimal coordinated control techniques were used in LES to increase the total wind farm power extraction. In [5], a simple distributed population-games-based algorithm was proposed for wind farm control with multiple estimated gradients being used. In [6], a wind farm controller was proposed to adjust the power generation of individual turbine to match the grid requirements. In [7], an optimal active power control method was developed to optimize the pitch angle and active power curves. In [8] and [9], a bi-level decentralized active and reactive power controller was designed for a large-scale wind farm cluster consisting of multiple wind farms. In [10], a constrained closed-loop wind farm controller was proposed to provide secondary frequency regulation and power tracking. In [11], a control strategy was proposed to allocate power regulation task to individual turbines to satisfy the overall dispatch order. In [12], a constrained MPC was proposed to minimize power losses due to wakes in a wind farm. In [13], a MPC based distributed coordinated active and reactive power control scheme for a wind farm was proposed. In [14], a nonlinear MPC scheme was proposed for a wind farm to achieve the objectives of both frequency response and wind generator stability.

However, most of the afore-mentioned control schemes were designed based on analytical wind farm models or LES study. The detailed turbulent wind flow state in the atmospheric boundary layer is very high-dimensional, and an accurate LES state model is computationally very expensive. Hence, the induced computational cost is rather intractable for real-time control operations. The above optimization methods were also typically based on the estimated gradients of cost functions, which may lead to local suboptimal solutions. Actually, the lack of suitable tools for modeling the interactions between the turbines and the flow across the wind farm is a major hurdle to effective farm level control. Such tools need to address multi-fidelity dynamic modelling and offer the right balance among simplicity and fidelity for developing closed-loop advanced control algorithms at a farm level.

As the latest paradigm in computational intelligence, the DNL has demonstrated greater potential over traditional machine learning methods and thus has attracted substantial attention [15]. It can model extremely sophisticated functions and can discover intricate structures from natural data in its raw forms through multiple levels of abstraction and non-linear processing layers trainable from the beginning to the end. In addition, it has dynamic nature and can deal with varying conditions within relatively small timescales, and hence is suitable for modelling wind farm flow dynamics.

This paper leverages the latest development of the DNL to address the dynamic modelling and predictive control of the offshore wind farm. As shown in Fig. 1, the offshore wind farm is connected to the main AC (alternating current) grid through a VSC-HVDC (voltage source converter - high voltage direct

X. Yin and X. Zhao (corresponding author) are with the School of Engineering, The University of Warwick, Coventry CV4 7AL, U.K. E-mails: x.yin.2@warwick.ac.uk, xiaowei.zhao@warwick.ac.uk. This work was funded by the UK Engineering and Physical Sciences Research Council under grant EP/R007470/1.

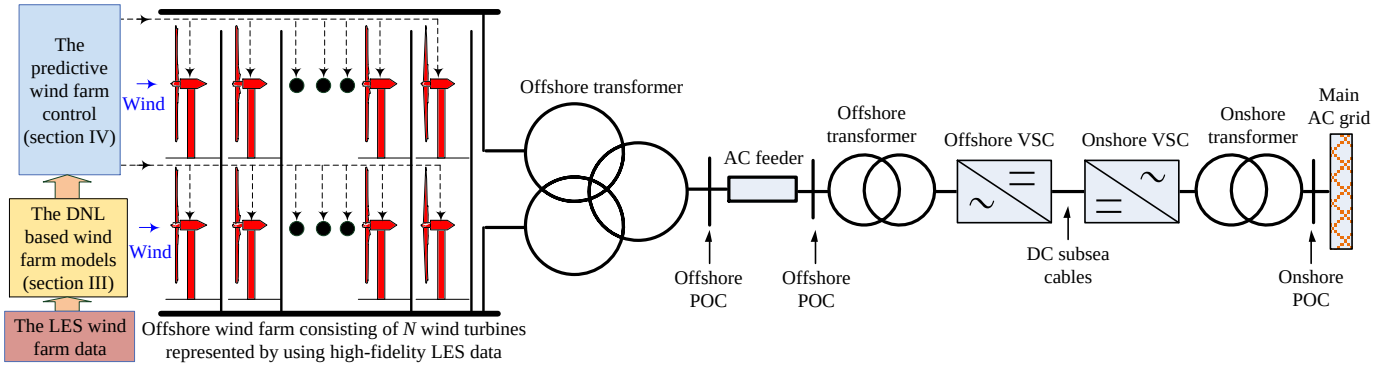

Fig. 1 The offshore wind farm connected to the main AC grid through VSC-HVDC link, where AC: alternating current, HVDC: high voltage direct current, POC: point of connection, VSC: voltage source converter, and • denotes other wind turbines that are not depicted.

current) transmission link that comprises the offshore transformers, AC feeder, VSCs, DC (Direct Current) subsea cables and onshore transformer. The paper focuses on the DNL based predictive control of the offshore wind farm that consists of *N* wind turbines represented by using the high-fidelity LES data. The DNL model to be established in Sec. III is a hybrid CNN-LSTM model that combines the CNN and LSTM to predict wind farm outputs by using the LES data. Based on this hybrid models, decentralized and distributed MPC methods (see Fig. 3) are designed in Sec. IV to maximize the power output of the wind farm while guaranteeing reliable operations.

The main contributions of the work are:

(a). The hybrid CNN-LSTM model with a deep learning architecture is established based on high-fidelity LES data to predict wind farm outputs with enhanced capabilities of processing strong spatial and temporal correlations of the LES data.

(b). Based on the established CNN-LSTM models, distributed and decentralized MPC methods are designed to maximize power generation of the wind farm while guaranteeing reliable operations.

(c). The feasibility and effectiveness of the CNN-LSTM based wind farm models and the MPC methods are evaluated through extensive computational simulations of two typical wind farm cases.

II. The Wind Farm and Wind Turbine Models

This section develops a dynamic wind farm model based on the LES data, and a dynamic wind turbine model.

A. The Dynamic Wind Farm Model

The wind farm flow dynamics can be generally modelled by the standard incompressible three-dimensional Navier–Stokes equations [16]. These equations are spatially discretized over a staggered grid by the hybrid differencing scheme and the finite volume method [17], where each wind turbine is modelled as a classical non-rotating actuator disk. The temporal discretization can be transformed to a two-dimensional difference algebraic equation (1) which retains the main elemental features of the three-dimensional turbulent wind flows [17].

$$E\big(q(k)\big)q(k+1) = Aq(k) + b\big(q(k), w(k)\big) \tag{1}$$

where $E\big(q(k)\big)$ is a non-singular square descriptor matrix containing the diffusion and convection terms after spatial discretization, $q(k)$ is a state vector including the longitudinal and lateral flow velocities and pressure variables along the grid points at the time step *k*, *A* is a constant matrix representing the temporal discretization of the flow depending on the chosen sampling time, $b\big(q(k), w(k)\big)$ contains the turbine forcing terms and boundary conditions, $w(k)$ is the vector of control variables including the disk-based thrust coefficient $C(k)$ and yaw angle $\gamma(k)$ at the time step *k*.

The above terms are described as

$$q(k) = \begin{bmatrix} u(k) \\ v(k) \\ p(k) \end{bmatrix}, w(k) = \begin{bmatrix} C(k) \\ \gamma(k) \end{bmatrix} C(k) = \begin{bmatrix} C'_{T1}(k) \\ C'_{T2}(k) \\ \vdots \\ C'_{TN}(k) \end{bmatrix}, \gamma(k) = \begin{bmatrix} \gamma_1(k) \\ \gamma_2(k) \\ \vdots \\ \gamma_N(k) \end{bmatrix} \tag{2}$$

where $u(k)$, $v(k)$ and $p(k)$ are respectively the longitudinal and lateral flow velocities and pressure at the time step *k*, $C'_{Tn}(k)$ is the disk-based thrust coefficient of turbine *n* at time step *k*, *N* is the number of wind turbines in the wind farm, $N \geqslant n$.

The wind farm model (1) is a large-scale system whose order is determined based on the chosen domain size and staggered grid resolution. It can represent the complex wake interactions and turbulence. By denoting the prediction of $x(k+i)$ by $x(k+i\,|\,k)$ provided the initial state $x(k)$ at time step *k*, Eq. (1) is expanded as

$$\begin{aligned} &E\big(q(k+1\,|\,k)\big)q(k+2\,|\,k) = Aq(k+1\,|\,k) + b\big(q(k+1\,|\,k), w(k+1\,|\,k)\big) \\ &= AE^{-1}\big(q(k\,|\,k)\big)Aq(k\,|\,k) + AE^{-1}\big(q(k\,|\,k)\big)b\big(q(k\,|\,k), w(k\,|\,k)\big) \\ &+b\big(q(k+1\,|\,k), w(k+1\,|\,k)\big) \end{aligned} \tag{3}$$

and

$$\begin{aligned} &E\big(q(k+2\,|\,k)\big)q(k+3\,|\,k) = Aq(k+2\,|\,k) + b\big(q(k+2\,|\,k), w(k+2\,|\,k)\big) \\ &= AE^{-1}\big(q(k+1\,|\,k)\big)AE^{-1}\big(q(k\,|\,k)\big)Aq(k\,|\,k) \\ &+AE^{-1}\big(q(k+1\,|\,k)\big)AE^{-1}\big(q(k\,|\,k)\big)b\big(q(k\,|\,k), w(k\,|\,k)\big) \\ &+AE^{-1}\big(q(k+1\,|\,k)\big)b\big(q(k+1\,|\,k), w(k+1\,|\,k)\big) \\ &+b\big(q(k+2\,|\,k), w(k+2\,|\,k)\big) \end{aligned} \tag{4}$$

For the N_p time step ahead prediction, (1) can be expanded as

$$\begin{aligned}&E\left(q(k+N_p-1|k)\right)q(k+N_p|k)=AE^{-1}\left(q(k+N_p-2|k)\right)\\&AE^{-1}\left(q(k+N_p-3|k)\right)...AE^{-1}\left(q(k+1|k)\right)AE^{-1}\left(q(k|k)\right)Aq(k|k)\\&+AE^{-1}\left(q(k+N_p-2|k)\right)...AE^{-1}\left(q(k|k)\right)b\left(q(k|k),w(k|k)\right)+...\\&+AE^{-1}\left(q(k+N_p-2|k)\right)...AE^{-1}\left(q(k+j|k)\right)b\left(q(k+j|k),w(k+j|k)\right)\\&+...+AE^{-1}\left(q(k+N_p-2|k)\right)b\left(q(k+N_p-2|k),w(k+N_p-2|k)\right)\\&+...+b\left(q(k+N_p-1|k),w(k+N_p-1|k)\right),1\le j\le N_p-3.\end{aligned} \tag{5}$$

The inflow wind velocity for the turbine *n* can be represented by using the state vector $q(k)$. Therefore,

$$\begin{aligned}&V_n(k)=\sqrt{u_n^2(k)+v_n^2(k)}\Rightarrow\\&V_n(k+N_p|k)=\sqrt{u_n^2(k+N_p|k)+v_n^2(k+N_p|k)}\end{aligned} \tag{6}$$

where V_n denotes the inflow wind velocity for the turbine *n*.

B. The Dynamic Wind Turbine Model

The wind power captured by the turbine *n* in the wind farm is

$$P_n(k+N_p|k)=\frac{\rho\pi R^2}{2}C_{pn}(k+N_p|k)\left[V_n(k+N_p|k)\cos\left(\gamma_n(k+N_p|k)\right)\right]^3 \tag{7}$$

where P_n and C_{pn} are respectively the captured wind power and power coefficient of the turbine *n*, the parameters ρ and R are respectively the air density and rotor radius (same for all the turbines). The coefficient C_{pn} determines the proportion of the available aerodynamic power that a turbine can capture, which is related to the disk-based thrust coefficient as follows

$$C_{pn}=\frac{64C'_{Tn}}{\left(C'_{Tn}+4\right)^3} \tag{8}$$

It can also be represented as [18]

$$C_{pn}\left(\lambda_n,\beta_n\right)=\left(0.44-0.0167\beta_n\right)\sin\left(\frac{\pi\left(\lambda_n-3\right)}{15-0.3\beta_n}\right)-0.00184\left(\lambda_n-3\right)\beta_n \tag{9}$$

where λ_n and β_n are respectively the tip speed ratio and pitch angle of the turbine *n*. the turbine *n's* tip speed ratio is

$$\lambda_n=\frac{\omega_{tn}R}{V_n}=\frac{\omega_{gn}R}{V_n i_{gn}} \tag{10}$$

where ω_{tn} and ω_{gn} are respectively the turbine rotor speed and generator speed, which are related through the constant gear transmission ratio i_{gn}.

The turbine torque is represented as

$$T_n=\frac{\pi\rho R^5\omega_{gn}^2}{2\lambda_n^3 i_{gn}^2}C_{pn}\left(\lambda_n,\beta_n\right) \tag{11}$$

where T_n is the rotation torque for the turbine *n*.

The turbine's drive-train dynamics are represented as [19]

$$\begin{aligned}&\frac{J_{tn}}{T_s}\left(\omega_{gn}(k+1)-\omega_{gn}(k)\right)=T_n(k)-k_t\omega_{gn}(k)-T_{gn}(k)i_{gn}\\&\Rightarrow\omega_{gn}(k+1)=\left(1-\frac{k_tT_s}{J_{tn}}\right)\omega_{gn}(k)+\frac{T_s}{J_{tn}}\left(T_n(k)-T_{gn}(k)i_{gn}\right)\end{aligned} \tag{12}$$

where k_t and J_{tn} are respectively the damping ratio and equivalent inertia of the turbine *n*, T_s denotes the sampling time interval, and T_{gn} denotes the generator torque control input.

By expanding (12) for N_p time step ahead, one obtains

$$\begin{aligned}&\omega_{gn}(k+N_p|k)=\left(1-\frac{k_tT_s}{J_{tn}}\right)^{N_p}\omega_{gn}(k|k)+\\&\left(1-\frac{k_tT_s}{J_{tn}}\right)^{N_p-1}\frac{T_s}{J_{tn}}\left(T_n(k|k)-T_{gn}(k|k)i_{gn}\right)+...\\&+\frac{T_s}{J_{tn}}\left(T_n(k+N_p-1|k)-T_{gn}(k+N_p-1|k)i_{gn}\right)\end{aligned} \tag{13}$$

Above the rated wind speed, the pitch control system acts to maintain the rotor speed at the rated value. Its dynamics can be represented in the discrete time form as

$$\beta_n(k+1)=\frac{T_s}{\tau_\beta}\left(\beta_{rn}(k)-\beta_n(k)\right)+\beta_n(k)=\frac{T_s}{\tau_\beta}\beta_{rn}(k)+\left(1-\frac{T_s}{\tau_\beta}\right)\beta_n(k) \tag{14}$$

where β_n, β_{rn} and τ_β are respectively the pitch angle, pitch control command and the time constant of the pitch system.

By expanding (14) for N_p time step ahead, one obtains

$$\begin{aligned}&\beta_n(k+N_p|k)=\left(1-\frac{T_s}{\tau_\beta}\right)^{N_p}\beta_n(k|k)+\left(1-\frac{T_s}{\tau_\beta}\right)^{N_p-1}\frac{T_s}{\tau_\beta}\beta_{rn}(k|k)\\&+...+\frac{T_s}{\tau_\beta}\beta_{rn}(k+N_p-1|k)\end{aligned} \tag{15}$$

By using the similar principle as the pitch system, the yaw mechanism dynamics can be expressed as

$$\begin{aligned}&\gamma_n(k+N_p|k)=\left(1-\frac{T_s}{\tau_\gamma}\right)^{N_p}\gamma_n(k|k)+\left(1-\frac{T_s}{\tau_\gamma}\right)^{N_p-1}\frac{T_s}{\tau_\gamma}\gamma_{rn}(k|k)+...\\&+\frac{T_s}{\tau_\gamma}\gamma_{rn}(k+N_p-1|k)\end{aligned} \tag{16}$$

where γ_n, γ_{rn} and τ_γ are respectively the yaw angle, yaw control command and the time constant of the yaw mechanism.

III. Deep Neural Learning Based Wind Farm Models

In this section, a hybrid CNN-LSTM model with a deep learning architecture is proposed for predicting wind speed, generator speed and the wind turbine power, which represents the farm dynamics following the relationships in section II.

A. The Hybrid CNN-LSTM Model

The CNN-LSTM model is a hybrid LSTM architecture particularly designed for sequence prediction. It involves CNN layers for feature extraction on input data and LSTM units to support sequence prediction. The CNN typically includes convolutional layers, pooling layers, hidden layers and fully connected layers. The convolutional layers are used with learning filters that represent features of the input and generate a feature map. The pooling layers perform non-linear down samplings by combining a cluster of neurons at one layer into the next single neuron based on non-linear functions including max pooling and average pooling. The fully connected layers

are then added after the convolutional and pooling layers for final output. The LSTM units are introduced to circumvent the vanishing and exploding gradient problems in the general recurrent neural network (RNN). The LSTM units use input gates, output gates, and forget gates to control modifying, accessing and storing of the internal states, and hence to discover long-range temporal relationships from the input sequences [20]. In order to alleviate the over-fitting problems, the "dropout" technique is employed, which randomly drops out some hidden and visible units to make nodes more insensitive to the weights of other nodes, and hence provides a method for approximately combining many different neural network architectures efficiently.

Fig. 2 describes the overall architecture of the employed hybrid CNN-LSTM model for wind farm predictions. This model contains seven layers with weights, which contains two convolutional layers with Max pooling after each, one LSTM layer, one dropout layer (not shown in Fig. 2), and one fully connected output layer for output predictions.

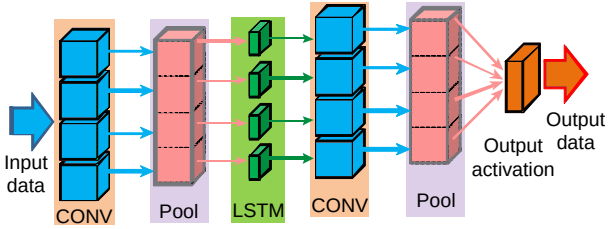

Fig. 2 The architecture of the CNN-LSTM model for wind farm predictions

The input data has been preprocessed in such a way that each data record contains 50 time slices. This results in a matrix of length 50. The first one-dimensional convolutional layer defines a filter with the kernel size of three, and totally 64 filters are defined, which allows to train 64 different features on the first layer. The output of the first layer is a $50 * 64$ neuron matrix. Each column of the output matrix holds the weights of one single filter. With the defined kernel size and considering the length of the input matrix, each filter will contain 50 weights. The first Max pooling layer is used after the first convolutional layer in order to prevent overfitting of the data and reduce the complexity of the output. The pool length of this layer is chosen as three, which means the size of the output matrix of this layer is only a third of the input matrix.

To learn the temporal structure from input sequences, the LSTM hidden layer is added after the first Max pooling layer, which forms the hybrid CNN-LSTM architecture. In this case, 200 LSTM units are defined in this hidden layer and each LSTM unit has connections with other nodes in LSTM layer. The LSTM units possess the ability to learn long range dependency from the input sequences and each LSTM unit essentially acts as an accumulator of the state information. Actually, each LSTM unit has a memory cell to store the internal state and three additional gates (input, output and forget gates) to control the behavior between the memory cell, the input and the output cells [21].

To reduce over-fitting, a dropout layer is added after the LSTM layer, which will randomly assign zero weights to the neurons in the network. A rate of 0.1 is chosen and 10% of the neurons will receive a zero weight. With this operation, the network becomes less sensitive to smaller variations in the data and further increase the accuracy on unseen data. The result from the dropout layer will be fed into the second convolutional layer and the second Max pooling layer with the length of 3. Then, 64 different filters are defined and trained on this convolutional layer level and the Max pooling layer is used to increase the richness of features and distill the filter maps down to the size that includes the most salient features.

The fully connected output layer is defined to recombine the representations learned by convolutional layer and reduces the dimension. This final layer will flatten down the neural network structure to a single one-dimensional vector by using matrix multiplication. The activation function ReLU is used to solve the problem of gradient explosion and speed up the forward propagation process.

The above defined hybrid model is trained and fitted on the training dataset using the efficient Adam version of stochastic gradient descent, and optimized using the mean squared error, or "mse" loss function. For efficient training, the learning rate is selected to be 0.001 and the truncated Back Propagation Through Time (BPTT) algorithm is used to compute gradients for the LSTM layer, which reduces the parameters and simplifies the complexity of the model.

The number of the total parameters of the hybrid CNN-LSTM model is a little greater than the traditional CNN architecture and can be viewed as deep architecture through time steps with the LSTM parts sharing the same parameters. Therefore, this hybrid architecture has more powerful representation ability than the traditional RNN and CNN.

B. The CNN-LSTM Based Wind Farm Model

The CNN-LSTM based wind farm models can be established by using the formulae in section II and the CNN-LSTM model in section III-A. As illustrated in (3)-(5), the flow field state variable $q(k+N_p \mid k)$ can be represented as a function f_q of the previous state variables and control input vector:

$$q(k+N_p \mid k) = f_q\begin{pmatrix} q(k \mid k), q(k+1 \mid k), ..., q(k+N_p-1 \mid k), \\ w(k \mid k), w(k+1 \mid k), ..., w(k+N_p-1 \mid k) \end{pmatrix} \tag{17}$$

The inflow wind speed $V_n(k+N_p \mid k)$ for the turbine n is a function of the inflow wind speeds and control inputs from the time step k to the time step $k+N_p$-1 of the turbines in front of it and its own.

By observing (11) and (13), the generator rotation speed of the turbine n can be described as

$$\omega_{gn}(k+N_p \mid k) = f_\omega\begin{pmatrix} \omega_{gn}(k \mid k), \omega_{gn}(k+1 \mid k), ..., \omega_{gn}(k+N_p-1 \mid k), \\ T_{gn}(k \mid k), T_{gn}(k+1 \mid k), ..., T_{gn}(k+N_p-1 \mid k), \\ C_{pn}(k \mid k), C_{pn}(k+1 \mid k), ..., C_{pn}(k+N_p-1 \mid k) \end{pmatrix} \tag{18}$$

Based on (15), the pitch angle for the turbine n at the $k+N_p$ time step can be represented as

$$\beta_n(k+N_p \mid k) = f_\beta\begin{pmatrix} \beta_n(k \mid k), \beta_{rn}(k \mid k), \beta_{rn}(k+1 \mid k), \\ ..., \beta_{rn}(k+N_p-1 \mid k) \end{pmatrix} \tag{19}$$

In the similar way, by using (16), the yaw angle for the turbine n at the time step $k+N_p$ is

$$\gamma_n(k+N_p\,|\,k)=f_\gamma\begin{pmatrix}\gamma_n(k\,|\,k),\gamma_{rn}(k\,|\,k),\gamma_{rn}(k+1\,|\,k),\\ ...,\gamma_{rn}(k+N_p-1\,|\,k)\end{pmatrix} \tag{20}$$

The Eqs. (19) and (20) indicate that the pitch and yaw angles are directly and respectively related with their previous states at the time step k and the control commands until the time step k+N_p-1.

By using (9), the power coefficient $C_{pn}(k+N_p\,|\,k)$ for the turbine n can be described as a function f_C of the inflow wind speed, the generator speed and the pitch angle of turbine n:

$$C_{pn}(k+N_p\,|\,k)=f_C\left(V_n(k+N_p\,|\,k),\omega_{gn}(k+N_p\,|\,k),\beta_n(k+N_p\,|\,k)\right) \tag{21}$$

Eq. (21) means that the power coefficient of the turbine n at the time step k+N_p is directly related with the wind speed, the generator speed and the pitch angle at the time step k+N_p.

By observing (8), it is obvious that the disk-based thrust coefficient $C'_{Tn}(k+N_p\,|\,k)$ at the time step k+N_p is also directly related with the wind speed, the generator speed and the pitch angle at the time step k+N_p.

By using (7), the wind power captured by the turbine n can be represented as a function f_P of the inflow wind speed, the generator speed, the pitch and yaw angles at the time step k+N_p:

$$P_n(k+N_p\,|\,k)=f_P\begin{pmatrix}V_n(k+N_p\,|\,k),\omega_{gn}(k+N_p\,|\,k)\\ \beta_n(k+N_p\,|\,k),\gamma_n(k+N_p\,|\,k)\end{pmatrix} \tag{22}$$

Based on (17)-(22), the CNN-LSTM models can be established to represent the dynamics of a wind farm. By using (17)-(21), the inflow wind speed $V_n(k+N_p\,|\,k)$ of the turbine n can be predicted by training a CNN-LSTM model using the inflow wind speeds, the generator speeds, the pitch and yaw angle control inputs from the time step k to the time step k+N_p-1 of the turbines in front of it and its own as the inputs. By using (18)-(21), the generator speed $\omega_{gn}(k+N_p\,|\,k)$ for the turbine n can be predicted by training a CNN-LSTM model for the generator speed by using the wind speeds, the generator speeds, the pitch and yaw angle control commands from the time step k to the time step k+N_p-1 of its own. Based on the above CNN-LSTM models, (21) and (22), the captured wind power $P_n(k+N_p\,|\,k)$ of the turbine n at the time step k+N_p can be predicted by training the third CNN-LSTM model that uses the inflow wind speeds, the generator speeds, the pitch and yaw angle control inputs from the time step k to the time step k+N_p-1 of the turbines in front of it and its own as the inputs.

IV. The Predictive Control Design for Wind Farm

In this section, based on the CNN-LSTM based wind farm models in section III, the distributed MPC is designed for maximizing the wind farm power generation and minimizing the usage of control commands. A decentralized MPC method is designed for comparison purpose.

A. Control Problem Formulation

The MPC is a receding horizon approach in which the control signals are optimized and solved for a future time window. In order to obtain the control signals at each time step, a quadratic cost function needs to be defined. The cost function for the turbine n to be minimized has been chosen in order to achieve the maximal output power and minimal actuator usage.

Therefore, the cost function for the turbine n is

$$J_n=\sum_{i=1}^{N_p}q_{1n}P_n(k+i\,|\,k)+\sum_{i=0}^{N_c}q_{2n}u_n^T(k+i\,|\,k)u_n(k+i\,|\,k) \tag{23}$$

where q_{1n} is a minus weight for maximizing the turbine power generation, q_{2n} is a weight for penalizing the actuator usage, N_p and N_c respectively denote the prediction horizon and control moves, and

$$u_n(k+i\,|\,k)=\left[T_{gn}(k+i\,|\,k),\beta_{rn}(k+i\,|\,k),\gamma_{rn}(k+i\,|\,k)\right]^T$$

denotes the control input vector at the time step k+i.

Based on (23), the cost function for maximizing the total wind farm power production and minimizing the usage of the control commands can be derived as

$$J=\sum_{n=1}^{N}q_{3n}J_n \tag{24}$$

where q_{3n} is the weight for the turbine n.

In addition, in order to guarantee the safe operation of the wind farm, the following constraints on the realistic control inputs are required to be satisfied.

$$\begin{cases}T_{gn\min}\le T_{gn}(k+i\,|\,k)\le T_{gn\max}\\ \beta_{rn\min}\le\beta_{rn}(k+i\,|\,k)\le\beta_{rn\max}\\ \gamma_{rn\min}\le\gamma_{rn}(k+i\,|\,k)\le\gamma_{rn\max}\\ i=0,1,...,N_c\end{cases} \tag{25}$$

$$\begin{cases}\Delta T_{gn\min}\le\Delta T_{gn}(k+i\,|\,k)\le\Delta T_{gn\max}\\ \Delta\beta_{rn\min}\le\Delta\beta_{rn}(k+i\,|\,k)\le\Delta\beta_{rn\max}\\ \Delta\gamma_{rn\min}\le\Delta\gamma_{rn}(k+i\,|\,k)\le\Delta\gamma_{rn\max}\\ i=0,1,...,N_c\end{cases} \tag{26}$$

where $T_{gn\min}$, $T_{gn\max}$, $\Delta T_{gn\min}$ and $\Delta T_{gn\max}$ are respectively the lower and upper limits of the generator torque and torque increment for the turbine n, $\beta_{rn\min}$, $\beta_{rn\max}$, $\Delta\beta_{rn\min}$, and $\Delta\beta_{rn\max}$ are respectively the lower and upper limits of the pitch angle control command and pitch control command increment for the turbine n, $\gamma_{rn\min}$, $\gamma_{rn\max}$, $\Delta\gamma_{rn\min}$, and $\Delta\gamma_{rn\max}$ are respectively the lower and upper limits of the yaw angle control command and yaw control command increment for the turbine n, $\Delta T_{gn}(k+i\,|\,k)$, $\Delta\beta_{rn}(k+i\,|\,k)$ and $\Delta\gamma_{rn}(k+i\,|\,k)$ are respectively the incremental values for the generator torque, the pitch control command and the yaw angle control command for the turbine n at the time step k+i, respectively.

It is obvious from (26) that the actuator's slew rates are limited at each control interval, which improves the reliability of the wind farm. Typically, the constraints in (25) and (26) result in more precise solutions of the control inputs.

B. The Predictive Control Design

The MPC exploits the developed CNN-LSTM based wind farm models to predict the future behavior of the variables to be controlled at each time interval. Its core idea is to solve a finite horizon optimal control problem online at each sampling time instant using the currently measured wind farm states and the CNN-LSTM based wind farm models, with the control

constraints in (25) and (26). The MPC is capable of handling multivariable constrained wind farm control problem and finding the optimal solution at each time interval by virtue of its online optimization nature. Hence, a decentralized MPC and a distributed MPC are designed for wind farm control. The decentralized MPC consists of N local MPC controllers. Based on the objective function in (23) and the trained CNN-LSTM based wind farm models in section III, each local MPC controller can be designed to control a wind turbine separately by resolving a constrained optimization problem online at each sampling period. Therefore,

$$\begin{cases} \min J_n \text{ in (23)} \\ \text{subject to (25), (26) and the three CNN-LSTM models with } n=1,2,...,N \end{cases} \tag{27}$$

The decentralized MPC represents a single wind turbine optimization approach in the wind farm and therefore can be used to distribute the computational burden to each local MPC controller. Its optimal control objective is to maximize the power generation of each individual wind turbine and limit the usage of its control actuators.

Rather than running individually like the decentralized MPC, the distributed MPC treats the entire wind farm as a comprehensive real-time optimization object and determines the control signals based on the entire wind farm topology. As a supervisory controller, wind turbines are coordinated and run iteratively to approximate the optimal operations set by the distributed MPC so that the optimal performance of the entire wind farm is achieved. This distributed MPC combines measurement data and takes into account the trade-off between wind energy production and the usage of actuators. By following the receding horizon control concept, this coordination-based distributed MPC can be defined as

$$\begin{cases} \min J = \sum_{n=1}^{N} q_{3n} J_n \\ \text{subject to (25), (26) and the three CNN-LSTM models with } n=1,2,...,N \end{cases} \tag{28}$$

The objective function (28) makes the distributed MPC developed for the wind farm different from the aforementioned decentralized MPC. It is a cooperative wind farm level closed-loop control paradigm.

In order to solve the control problems in (27) and (28), the differential evolution (DE) algorithm is used, which is selected due to the complexity of the trained CNN-LSTM wind farm models and the considered objective functions in (27) and (28). As a stochastic optimization algorithm, the DE algorithm does not rely on the traditional gradient descent method for convergence and can therefore find the global optimal solutions with high efficiency [22]. In comparison with the evolutionary optimization algorithms such as genetic algorithm or particle swarm, the DE algorithm has fast convergence speed that is suitable and efficient for optimizing large-scale wind farms in an iterative way. After the optimization at each time interval, an optimal control sequence can be subsequently obtained in each finite horizon and the first step of the optimal control sequence is applied and then the optimization procedure repeats itself.

The above DNL-based distributed MPC is described by the block diagram in Fig. 3.

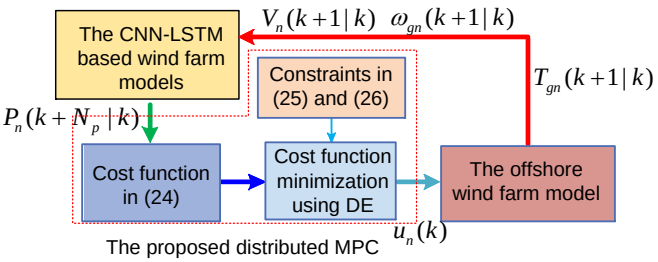

Fig. 3 Block diagram of the proposed distributed MPC

V. Case Study and Validations

This section verify the feasibility and effectiveness of the CNN-LSTM based wind farm models and the two MPC methods by simulation studies with high fidelity LES data.

A. The Data Preparation

The original data for training the CNN-LSTM models and designing the predictive control methods are obtained from the high-fidelity simulations from the Simulator fOr Wind Farm Applications (SOWFA), developed by the National Renewable Energy Laboratory (NREL). The simulations are run based on a two-turbine and a nine-turbine wind farm models, respectively. The SOWFA solves the filtered, unsteady, three-dimensional, incompressible Navier–Stokes equations over a finite spatial and temporal mesh, accounting for the geostrophic and Coriolis forcing terms, and can provide accurate and high-fidelity simulation data of turbulent atmospheric flows together with the analysis of wind farm and wind turbine fluid physics and structural response at a fraction of the cost of field tests. As a LES solver, the SOWFA was designed based upon the coupling of the OpenFOAM Computational Fluid Dynamics (CFD) tool-kit [23] with the high- fidelity NREL's aero-elastic wind turbine simulation tool OpenFAST. Each wind turbine in the SOWFA is designed as the NREL's 5 MW reference wind turbine and is represented using the actuator line model coupled with the OpenFAST to address the flow interactions. The aerodynamics or the "momentum" part of each turbine model is replaced by FAST inflow information at blade elements. The interested readers are referred to [24] for more details.

The wind farms are simulated with the turbulence closure and the atmospheric boundary layer for the wind flow solution. In addition, different from onshore wind farms, offshore wind farms are greatly influenced by atmospheric thermal conditions and complex sea states. In the SOWFA, the thermal effects are considered through the precursor simulation of atmosphere boundary layer while the sea states are considered via the sea surface roughness estimation. The consideration of these properties makes SOWFA a suitable tool in simulating offshore wind farms operating under realistic conditions. The simulated LES data are able to capture the dominant dynamics of the offshore wind farms including the wake interactions.

In the chosen simulation scenarios, an 8 m/s inflow wind speed from the left along the x axis is set as the wind flow input for the wind farm (Figs. 7 and 12). The turbines are operated with generator torque control for maximizing the wind farm power production. The simulation results from the SOWFA are then employed to establish the CNN-LSTM models in section

III-B. The number of the CNN-LSTM wind farm models is decided based on the prediction horizon N_p and the control moves N_c, where N_c is decided as N_p-1, and N_p is chosen as 3 for three-step look ahead prediction. The CNN-LSTM models are designed and trained in Keras with TensorFlow backend. The architecture of the used CNN-LSTM models is designed based on Fig. 2 and contains seven layers followed by a Dense layer on the output. The CNN-LSTM models are trained to build up internal states and update weights using the BPTT with a batch size of 78 across the internal vector representations of input sequences described in section III-B. The training data are normalized and rescaled into the range of [0, 1] before training, which stabilizes and speeds up the DNL training using gradient descent.

B. The Deep Neural Learning Results

The accuracy of the trained CNN-LSTM models has been tested against the real values and the mean absolute percentage error (MAPE) and the root mean square error (RMSE) are chosen to evaluate the prediction performances of these models. The MAPE and RMSE are defined respectively as

$$\mathrm{MAPE}(y_i)=\frac{1}{M}\sum_{i=1}^{M}\left|\frac{y_i-\hat{y}_i}{y_i}\right|\times 100\% \tag{29}$$

$$\mathrm{RMSE}(y_i)=\left(\frac{1}{M}\sum_{i=1}^{M}\left|y_i-\hat{y}_i\right|^2\right)^{1/2} \tag{30}$$

where y_i is the real value, $\hat{y}_i$ is the predicted value and M is the total number of data samples.

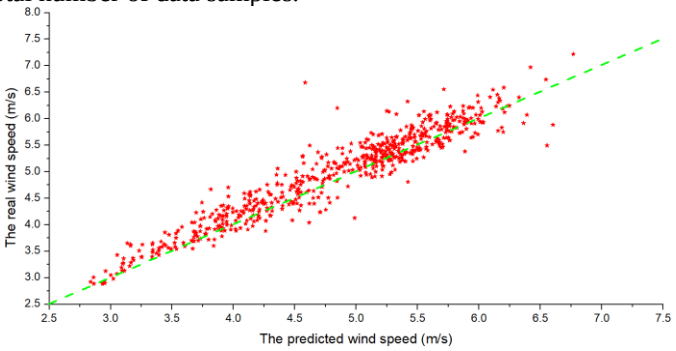

Fig. 4 The prediction results of the wind speed

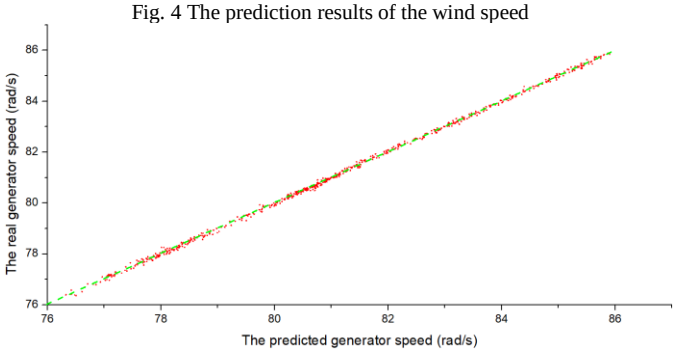

Fig. 5 The prediction results of the generator speed

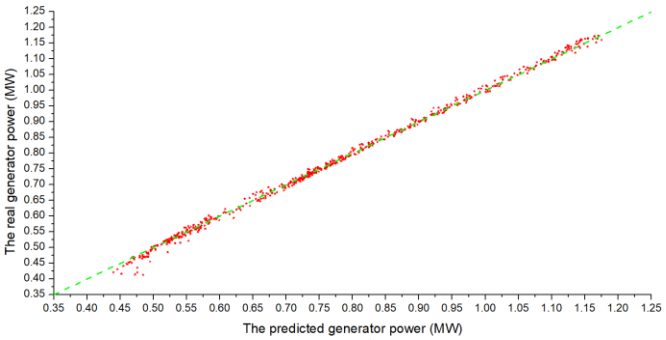

Fig. 6 The prediction results of the generator power

The test results of these CNN-LSTM models for predicting the inflow wind speed, the generator speed and power of the rear turbines in the wind farms (Turbine 2 in Fig. 7 and Turbine 8 in Fig. 12) are shown in Figs. 4-6, as representatives of the results from these models. The scattered red points in the figures are used to represent the test results. The *x* coordinates of these points denote the predicted values while their *y* coordinates represent the corresponding actual or real values. The dashed green diagonal lines are used to represent the exact match between the actual and predicted values. As the figures show, the scattered points are obviously aligned with the diagonal lines, which means that the predicted values of the inflow wind speed, generator speed and power are in very good agreement with their real values. Hence, the good prediction accuracy of the used CNN-LSTM prediction models is achieved. The prediction accuracy of the inflow wind speed seems to be a bit lower than the other two, which may be attributed to the relatively large dimension of the input data into the CNN-LSTM models of the wind speed prediction. The mean values of the MAPE for the inflow wind speed, generator speed and power predictions of the two turbines are respectively 2.91%, 0.4432% and 1.517%, and the corresponding mean values of the RMSE are respectively 0.205, 0.46 and 0.02238. All of these MAPE and RMSE values indicate that the relative prediction accuracy (calculated by 100%-MAPE) of the trained CNN-LSTM models can reach 97% or more, which is sufficiently accurate for predicting the necessary outputs for the wind farm control.

C. The Predictive Control Results

The two wind farm cases are used to verify the effectiveness of the predictive control approaches.

1) Case I: two-turbine wind farm

As shown in Fig. 7, the *x* coordinates of the turbines 1 and 2 are respectively 400 m and 1032 m, and the *y* coordinates of the two turbines are respectively 400 m and 400.096 m.

In the MPC design, for each wind turbine, 9 CNN-LSTM prediction models are established including the inflow wind speed, the generator speed and the generator power at 1st, 2nd and 3rd time step ahead, and there are totally 18 CNN-LSTM prediction models for the two-turbine case.

The two predictive control methods are designed based on the trained 18 CNN-LSTM models and their definitions in (27) and (28). The weights for the cost functions in (23) and (24) are chosen to be q_{11}=q_{12}=-60, q_{21}=q_{22}=0.01, q_{31}=1, q_{32}=0.5. The decision variables are chosen to be the generator torques of the two turbines, and 10 time steps are used to capture the dominant dynamic wake interactions of the wind farm. The DE algorithm is used to obtain the optimal solutions for the two control methods at each time interval.

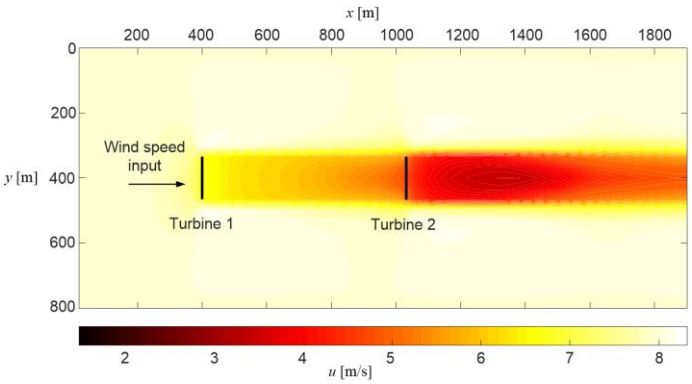

Fig. 7 The offshore two-turbine wind farm configuration using the SOWFA

Based on the trained CNN-LSTM models, the two MPC methods have been implemented and verified individually. As shown in Figs. 8 and 9, the distributed MPC has obviously outperformed the decentralized MPC in improving the wind farm power generation. By using the distributed MPC, the averaged wind farm power (defined as the total wind farm power divided by the number of wind turbines in the wind farm) can be increased by 8% to 30% in comparison with the decentralized MPC. The averaged and maximum increase rates are respectively 18.8% and 30%. The results demonstrate that the distributed MPC is clearly more effective in maximizing wind farm power generation than the decentralized MPC that is more "greedy" due to the use of the cost function (23) for individual wind turbine while not considering the power optimization at the whole wind farm level.

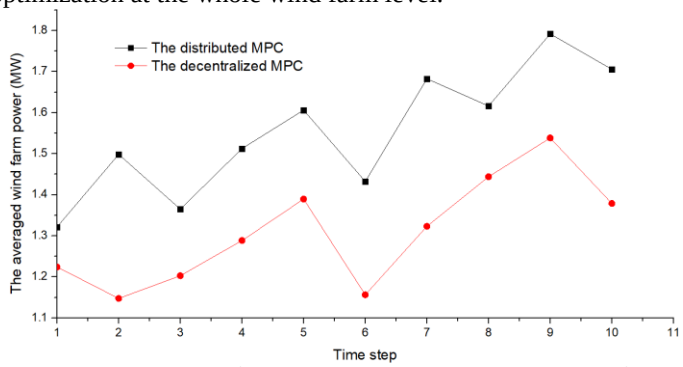

Fig. 8 The averaged wind farm power generations based on two types of the MPC controls

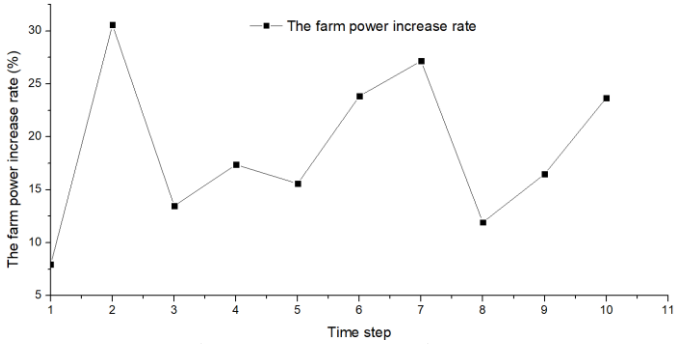

Fig. 9 The wind farm power increase rate of the distributed MPC

As illustrated in Fig. 10, the generator torques from the two MPC control methods for the front turbine (turbine 1) vary around 20 kNm, while they vary between 6 kNm and 18 kNm for the rear wind turbine (turbine 2). This is because that the front wind turbine has much higher inflow wind speed and the generator torque needs relatively large response to adapt to the changing wind speed and hence to capture the maximum wind power. Also, as can be seen from the figure, the generator torques from the distributed MPC method generally have smaller variation rates than the decentralized MPC method, which indicates that the distributed MPC has the potential to lead to more reliable control solutions. The computation time at each time step using the distributed MPC has also been calculated by using conventional single core computation and multi-core in parallel computation.

As shown in Fig. 11, comparing with the conventional computation method, the execution time at each time step is significantly reduced by around half due to the use of the multiple CPUs. Considering that the sampling time interval of the wind farm is around 5 s or longer, the computational time of around 1s or 0.7s is sufficiently fast enough to generate a timely control solution within a sampling time interval and therefore can guarantee safe and reliable wind farm operations.

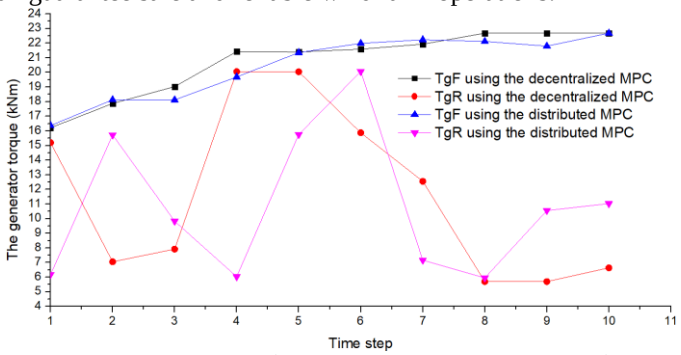

Fig. 10 The generator torques of the two turbines using the two types of MPC controls, TgF and TgR denote the generator torques of the front (turbine 1) and rear (turbine 2) wind turbines

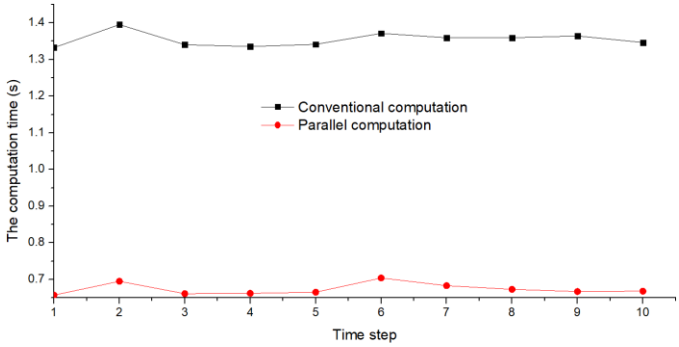

Fig. 11 The computational time at each time step using the distributed MPC

2) Case II: nine-turbine wind farm

A nine-turbine wind farm case is studied to further verify the effectiveness of the two MPC methods since the nine-turbine wind farm has more complex wake interactions and thus can provide a more in-depth insight into the wind farm control than the two-turbine wind farm case.

As shown in Fig. 12, the *x*-coordinates of turbines in the simulation grid are designed as [0.4048, 0.4024, 0.40, 1.0368, 1.0344, 1.0320, 1.6688, 1.6663, 1.6639] $\times 10^3$ m, and their *y*-coordinates are designed as [1.1584, 0.7792, 0.40, 1.1543, 0.7752, 0.3960, 1.1503, 0.7711, 0.3919] $\times 10^3$ m. The wind speed input of the wind farm is also 8 m/s.

For each wind turbine, 9 CNN-LSTM prediction models including wind speed, generator speed and power are established for the MPC design. Hence, 81 CNN-LSTM prediction models are used for the 9-turbine wind farm case.

The two MPC methods are designed based on the trained 81 CNN-LSTM models (27) and (28). The weights for the cost functions in (23) and (24) are chosen to be q_{1n} =-50, q_{2n}=0.02, (n=1, 2, .., 9), q_{3i}=1, q_{3j}=0.5, q_{3k}=0.25, (i=1, 2, 3, j=4, 5, 6, k=7, 8, 9). The turbine generator torques are used as the decision variables in the 10 time-step simulation. The optimal solutions for the two control methods at each time interval are also obtained by using the DE algorithm.

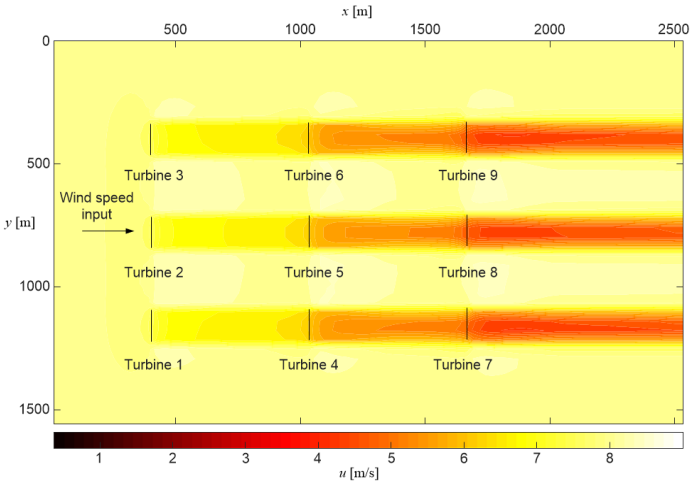

Fig. 12 The nine-turbine offshore wind farm configuration using the SOWFA

The nine-turbine wind farm has been simulated with the LES data based on the two MPC methods. The simulation results are presented in Figs. 13 and 14. As the figures show, the distributed MPC is clearly more capable of maximizing the wind farm power production than the decentralized MPC. In comparison with the decentralized MPC, the averaged wind farm power increases by 5% to 38% when the distributed MPC method is used. As illustrated in Fig. 14, the averaged power increase rate is 17% and the maximum increase rate is 38% when the distributed MPC method is applied. The better performance of distributed MPC is attributed to the fact that it is a cooperative farm-level control paradigm that coordinates all the wind turbines to maximize the total power generation at farm scale, while the decentralized MPC represents a single wind turbine optimization approach that aims to maximize the power generation of each individual wind turbine. The latter does not necessarily lead to the maximal total power generation at farm scale due to the complicated interactions (e.g., wakes) between the wind turbines. Thus here the former results in up to 38% increase on power generation at farm scale compared with the latter. The test results in Figs. 13 and 14 are also in good agreement with that in Figs. 8 and 9, which further demonstrates the effectiveness of the distributed MPC in improving the wind farm power generations.

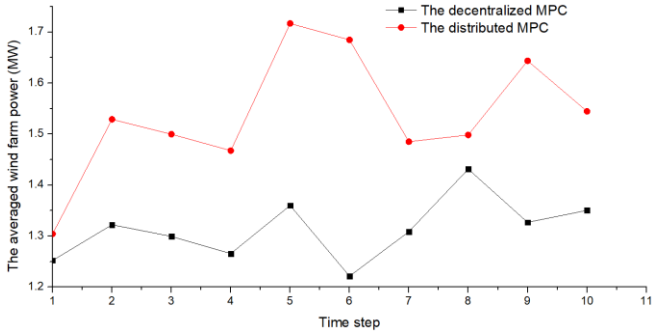

Fig. 13 The averaged wind farm power generations using the MPC methods

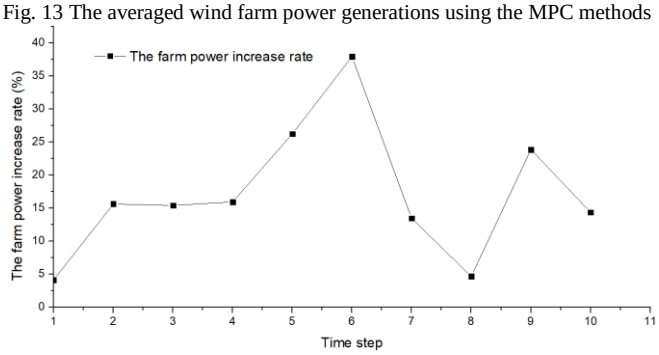

Fig. 14 The wind farm power increase rate of the distributed MPC

VI. Conclusion

The paper has explored the DNL based MPC for offshore wind farm using high fidelity LES data. The DNL architecture has been designed as the hybrid CNN-LSTM models which combine the speed of CNNs with the order sensitivity of the LSTM. The hybrid models can leverage the local and dense property from convolution operation and learn the temporal structure by storing information in the LSTM units, and hence are particularly suitable for closed-loop wind farm control as it is dynamic, including pitch, yaw and generator torque control capabilities, and handling temporally and spatially varying wind inflows. Then, the distributed and decentralized MPC methods were developed to resolve constrained optimal control problems of the wind farm power generation based on the trained CNN-LSTM models which considered the wake

coupling interactions among the turbines. Extensive simulations have been conducted to evaluate the accuracy and effectiveness of the hybrid models and the MPC methods by using high fidelity LES data through a two-turbine and a nine-turbine wind farm cases. The test results show that the trained CNN-LSTM models achieved a prediction accuracy of more than 97%. Compared with the decentralized MPC, an increase of up to 38% power generation has been achieved by using the distributed MPC. In addition, the computational efficiency of the distributed MPC is high enough to be applicable in real-time wind farm operations.

REFERENCES

[1] www.carbontrust.com/media/42162/ctc743-offshore-wind-power.pdf

[2] Park J, Law K H. A data-driven, cooperative wind farm control to maximize the total power production. Applied Energy, 2016, 165: 151-165.

[3] Ebrahimi F M, Khayatiyan A, Farjah E. A novel optimizing power control strategy for centralized wind farm control system. Renewable energy, 2016, 86: 399-408.

[4] Goit J, Munters W, Meyers J. Optimal coordinated control of power extraction in LES of a wind farm with entrance effects. Energies, 2016, 9(1): 29.

[5] Barreiro-Gomez J, Ocampo-Martinez C, Bianchi F D, et al. Data-Driven Decentralized Algorithm for Wind Farm Control with Population-Games Assistance. Energies, 2019, 12(6): 1164.

[6] Hur S, Leithead W E. Adjustment of wind farm power output through flexible turbine operation using wind farm control. Wind Energy, 2016, 19(9): 1667-1686.

[7] Tian J, Zhou D, Su C, et al. Optimal control to increase energy production of wind farm considering wake effect and lifetime estimation. Applied Sciences, 2017, 7(1): 65.

[8] Huang S, Wu Q, Guo Y, et al. Bi-level decentralised active power control for large-scale wind farm cluster. IET Renewable Power Generation, 2018, 12(13): 1486-1492.

[9] Huang S, Wu Q, Guo Y, et al. Bi-level decentralized active and reactive power control for large-scale wind farm cluster. International Journal of Electrical Power & Energy Systems, 2019, 111: 201-215.

[10] Boersma S, Doekemeijer B M, Siniscalchi-Minna S, et al. A constrained wind farm controller providing secondary frequency regulation: An LES study. Renewable energy, 2019, 134: 639-652.

[11] Lyu X, Jia Y, Xu Z. A Novel Control Strategy for Wind Farm Active Power Regulation Considering Wake Interaction. IEEE Transactions on Sustainable Energy, 2019.

[12] Vali M, Petrović V, Boersma S, et al. Adjoint-based model predictive control for optimal energy extraction in waked wind farms. Control Engineering Practice, 2019, 84: 48-62.

[13] Guo Y, Gao H, Wu Q, et al. Distributed coordinated active and reactive power control of wind farms based on model predictive control. International Journal of Electrical Power & Energy Systems, 2019, 104: 78-88.

[14] Kou P, Liang D, Linbo Y, et al. Nonlinear Model Predictive Control of Wind Farm for System Frequency Support. IEEE Transactions on Power Systems, 2019.

[15] Chen Q, Wang W, Wu F, et al. A Survey on an Emerging Area: Deep Learning for Smart City Data. IEEE Transactions on Emerging Topics in Computational Intelligence, 2019.

[16] Boersma S, Doekemeijer B, Vali M, et al. A control-oriented dynamic wind farm model: WFSim. Wind Energy Science, 2018, 3(1): 75-95.

[17] Versteeg H K, Malalasekera W. An introduction to computational fluid dynamics: the finite volume method. Pearson education, 2007.

[18] Kassem A M. Robust voltage control of a stand alone wind energy conversion system based on functional model predictive approach. International Journal of Electrical Power & Energy Systems, 2012, 41(1): 124-132.

[19] Asl H J, Yoon J. Power capture optimization of variable-speed wind turbines using an output feedback controller. Renewable Energy, 2016, 86: 517-525.

[20] C. Olah, Understanding LSTM Networks. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[21] https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

[22] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art. IEEE transactions on evolutionary computation, 2011, 15(1): 4-31.

[23] Jasak H, Jemcov A, Tukovic Z. OpenFOAM: A C++ library for complex physics simulations. International workshop on coupled methods in numerical dynamics. IUC Dubrovnik Croatia, 2007, 1000: 1-20.

[24] https://nwtc.nrel.gov/SOWFA.

Xiuxing Yin received his Ph. D. in mechatronic engineering from Zhejiang University, Hangzhou, China in 2016. He is now a research fellow at the University of Warwick, U. K. His research interests focus on mechatronics, renewable energy and nonlinear control.

Xiaowei Zhao is Professor of Control Engineering at the School of Engineering, University of Warwick. He obtained his PhD degree in Control Theory from Imperial College London. After that he worked as a postdoctoral researcher at the University of Oxford for three years before joining Warwick in 2013. His research interests include (1) control of wind/tidal turbines/farms; (2) grid integration of renewable energy; (3) microgrid; (4) control of fluid-structure interaction with applications to large and flexible wind turbines, highly flexible aircraft, and long-span suspension bridges; (5) control of coupled infinite-dimensional systems.